Contextualising Implicit Representations for Semantic Tasks

Theo W. Costain
Active Vision Lab
University of Oxford
costain@robots.ox.ac.uk

Kejie Li
Active Vision Lab
University of Oxford
kejie@robots.ox.ac.uk

Victor A. Prisacariu
Active Vision Lab
University of Oxford
victor@robots.ox.ac.uk

Abstract

Prior works have demonstrated that implicit representations trained only for reconstruction tasks typically generate encodings that are not useful for semantic tasks. In this work, we propose a method that contextualises the encodings of implicit representations, enabling their use in downstream tasks (e.g. semantic segmentation), without requiring access to the original training data or encoding network. Using an implicit representation trained for a reconstruction task alone, our contextualising module takes an encoding trained for reconstruction only and reveals meaningful semantic information that is hidden in the encodings, without compromising the reconstruction performance. With our proposed module, it becomes possible to pre-train implicit representations on larger datasets, improving their reconstruction performance compared to training on only a smaller labelled dataset, whilst maintaining their segmentation performance on the labelled dataset. Importantly, our method allows for future foundation implicit representation models to be fine-tuned on unseen tasks, without retraining the encoder.

1. Introduction

The explosion of interest in augmented reality in recent years has spurred a renewed search for more efficient representations of 3D data. Whilst point-clouds, meshes, and various other representations have been proposed over the years, the recent introduction of implicit representations like NeRF and DeepSDF have reignited interest in the *representation* rather than the processing of the data.

Opposed to "classical" representations that discretise the underlying structure, implicit representations (IRs) learn a continuous function over 3D space. IRs are able to represent the structure at arbitrary resolutions, trading spatial complexity for time complexity required to extract the structure from the representation. In the most current approaches, an encoder takes input in one or more modalities producing an encoding that is used to condition an MLP that composes the function. Early works, such as DeepSDF [35] and Occupancy Networks [31], learned functions that separated space into "inside" and "outside" regions, however, this ensured that the network could only learn closed surfaces. Subsequently, methods were proposed that resolved this limitation by learning an unsigned distance function (UDF), where the surface of the object lies on the zero level set of the function. More work followed, and improved on various aspects of these approaches including training ambiguities [55, 59] and extraction/rendering [59] (further discussion in Sec. 2), but despite this, relatively little attention was given to applications of these approaches in conventional pipelines, such as semantic segmentation or classification.

Foundation models, *i.e.* large pre-trained generalist networks and models (*e.g.* [16, 34]), that are trained on vast amounts of data, allowing adaptation to a variety of downstream tasks, are increasingly an essential building block in deep learning pipelines. It is not impossible that (as is already beginning to happen [34]) foundation like IR encoders may not be feasible or possible for most users to train from scratch, or even fine-tune encoding network, especially when the original training data is available (as is increasingly less common). Given Costain and Prisacariu [8] demonstrated that, when trained for reconstruction tasks only, IRs learn encodings that are not necessarily meaningful for semantic tasks. Accordingly, without the ability to train (or fine-tune) the encoder with semantic supervision [50], the performance on semantic tasks, using these encodings is likely to be unsatisfactory.

To address this problem, we propose a novel method to contextualise the encodings learnt by networks supervised on reconstruction tasks alone, even when the original reconstruction training data (*i.e.* ground truth distance function information) is not available. Basing our experiments on the approach of Wang et al. [50], we show the semantic limitations of encodings generated by training on reconstruction tasks alone. Then we propose our lightweight contextualising module that takes the learnt encoding and produces a small additional context encoding. This context encoding can then be combined with the existing encoding allowing the network to completely recover performance on the semantic tasks.

By separating the geometric tasks from the semantic tasks, our approach allows the geometric pipeline to be trained on much cheaper to produce datasets where complete semantic labels are not available, before our contextualising module is applied to a smaller fully labelled dataset enabling semantic segmentation alongside reconstruction. Rather than complex approaches [26], our method presents a simple, but effective and performant approach that address a major shortfall in existing implicit representation approaches.

Our key contributions are:

- Our contextualising module which reveals hidden semantic information contained in the feature encodings of IR.
- A novel and simple approach to train existing implicit representations for unseen semantic tasks without access to the original training data.

In the rest of this paper we cover: relevant existing works in the literature Section 2, our method and contextualising module Section 3, details of our experimental setup Section 4, the results of our experiments Section 5, and finally the limitations of our approach Section 6.

2. Related Work

Early IR works [2, 5, 13, 31, 32, 35, 38] focused mainly on reconstructing single objects. Both Occupancy Networks [31] and IM-Net [5], learn a function mapping from points in space to the probability that point lies within the object to be reconstructed. Occupancy Networks further proposed a hierarchical, octree based, extraction method to efficiently extract the mesh. In contrast, DeepSDF [35] learns a function mapping from space to a signed distance function. Although they proposed an encoder-decoder structure, they also introduced an auto-decoder structure, where the representation encoding is found by freezing decoder and optimising the encoding/embedding. Scene Representation Networks (SRN) [44] proposed a “Neural Renderer” module, which maps from 3D world coordinates to a feature representation of the scene at that location. Sign Agnostic Learning (SAL) [2] proposed to remove the need for signed ground truth information, whilst still learning a signed distance function. Crucial to this effort is an initialisation scheme that the initial level set was approximately a sphere of some chosen radius. Gropp et al. [13] introduce the Eikonal Loss term amongst other improvements to the loss function from SAL [2]. These new terms encourage the representation to develop a unit norm gradient, like a metric SDF, and acts as a geometric regularisation over the learned function, improving smoothness and accuracy of the reconstructions. Later methods [29] include approaches to better allow networks to represent high frequency information [45, 46], the former of which is vital to the performance of NeRFs [4, 30, 33].

These early works focused on single object reconstruction, with typically a single encoding or embedding per object. This limits the scale of objects these representations could represent, a concern later works proposed several solutions to. Many works arrived at a similar solution to this problem, using either planes [37] or grids [3, 6, 21, 37], to improve both the scale and detail of the reconstructions. Convolutional Occupancy Networks [37] make use of planes or grids of features, and IF-Net [6] learns learns a hierarchy of multi-scale features, both interpolating between these features at queried locations to predict occupancy probabilities and signed distance function respectively. Rather than interpolating features, Deep Local Shapes [3] and Jiang et al. [21], learn a grid of encodings, dividing scenes into small simple geometric shapes.

All the above methods share a common trait in separating space into inside *vs*. outside, however in the case where watertight meshes are not available (as is the case for common 3D Datasets [1, 9, 19]) training is not possible without complicated pre-processing, or learning overly thick walls. Chibane et al. [7] addressed this issue by learning an UDF as well as proposing a gradient based rendering scheme to extract the surface, a requirement given Marching Cubes [28] cannot be applied to UDFs. Various works followed in this vein [14, 47, 50, 55, 56, 59]. Notably, Guillard et al. [14], Zhou et al. [59] who independently proposed an approach to significantly improve the extraction/rendering of UDFs, by modifying Marching Cubes to look for diverging gradients rather than zero crossings allowing its use on UDFs.

A number of works have considered semantic tasks alongside NeRFs [24, 49, 51, 54, 58], however far fewer works [8, 23, 29, 50] consider semantic tasks alongside IRs. Costain and Prisacariu [8] argued that training IRs on geometric tasks alone produce encodings that are poor for semantic tasks. However, Luigi et al. [29] show that these encodings still contain the semantic information, and that it is possible to transform these encodings into a form that are more meaningful for semantic tasks. We leverage this insight in designing our contextualising module. Wang et al. [50], as well as proposing a UDF based IR, train a “surface-aware” segmentation branch alongside the UDF.

As a fundamental problem in computer vision, a vast array of works [10, 12, 15, 17, 18, 20, 25, 27, 39–43, 48, 52, 53, 57] have tackled semantic segmentation of point clouds, however a detailed discussion of these methods falls outside the scope of this work.

3. Method

Implicit representations seek to learn a functional mapping, f, from a query point, $q \in \mathbb{R}^3$, in space to the distance from that query point to the nearest point on the surface being represented. In this work we consider UDFs, further constraining $f : q \in \mathbb{R}^3 \mapsto \mathbb{R}_0^+$. In this work, we use UDFs, as these are the currently preferred way to represent non watertight scenes, however, this should not affect the generality of the approach, and should a dataset containing large watertight scenes be released, we expect our method should also apply. This function is typically implemented as a simple MLP, but to avoid overfitting the MLP to every surface to be represented, it is often desirable to condition the function on some global [31, 35] or local [7, 50] encoding of shape, giving $f : q \in \mathbb{R}^3, E \in \mathbb{R}^d \mapsto \mathbb{R}_0^+$, where E is some encoding vector and d its dimension.

As the only method that performs semantic tasks alongside learning implicit representations for large scenes, we use the RangeUDF method proposed by Wang et al. [50] as our baseline for our work. Their approach takes a sparse input point-cloud $P \in \mathbb{R}^{N\times3}$, where N is the number of points, and uses an encoder to learn some encoded features, $E_g \in \mathbb{R}^{N\times d}$. These encoded features are then passed to the decoder(s), alongside a set of query points $Q \in \mathbb{R}^{M\times3}$ (where M is the number of query points). KNN is used to collect the K nearest encoding vectors for each query point $q \in Q$, which are then combined using a simple attention module. These combined features, alongside the corresponding query point, are then fed into the UDF and semantic segmentation modules.

3.1. The Problem

A common pipeline in computer vision tasks is to take a pre-trained model, that produces meaningful features for a given task, and either fine-tune it, or use the generated features as input to another module that performs some desired task. The arc of research so far has resulted in this pre-training often [16] taking the form of classification tasks on extremely large datasets [11], ensuring these pre-trained models learn features that are semantically meaningful.

On the other hand, IR methods have arisen to tackle a different challenge: the representation of 3D shape and structure. Typically, this is in service of reducing the memory required to represent a given scene or object at high resolutions [31, 35, 37], compared to other conventional representations such as point-clouds, meshes, or voxel grids. Whilst much of the research on implicit representations to date has focused on the reconstruction task alone, there has been little consideration of how IRs might be used to replace conventional representations in existing pipelines, such as performing classification or segmentation, and other works [8] have shown that encodings learnt for reconstruction alone can provide insufficient information for semantic tasks.

When training the encodings that condition a UDF, the desire is for the network to learn some set of encodings E that holistically represent local structural information about the underlying shape. Our experiments at the beginning of Sec. 5, confirm [8] that the encodings, E_g, learnt when training the UDF for geometric reconstruction alone, show poor separability in semantic space (Fig. 2a). Whilst this can obviously be addressed by training for both semantic and reconstruction tasks jointly [8, 50], it is trivial to imagine scenarios where it is extremely desirable to be able to fine-tune on semantic tasks, without requiring either access to the original training data (original training data may not be publicly available), or having to potentially expensively retrain the *entire* pipeline. To address this, our contextualising module allows for the *effective* training of segmentation despite fixing the encoder/encodings. It also bears noting that whilst it is possible to exhaustively render/extract each scene from the encoding and UDF, and use this to re-create the training data, current implicit representation methods are far from perfect, and so taking this approach would almost certainly compound errors (akin to repeated photocopying of a paper document), not to mention the substantial cost of labelling the extracted scenes. Our proposed method avoids this entire problem, with a simple process.

3.2. Contextualising Module

The results of Zhou et al. [59] suggest that although not necessarily in present in a separable form, the semantic information is still present in the representation. Accordingly, we propose our simple contextualising module, which produces *compact* context features that carry substantial semantic information (Fig. 2b), that when combined with the original encoded features, produces features useful for semantic segmentation as well as reconstruction. An overview of our method is presented in Fig. 1.

Taking the encoded features, our module uses a small encoder-decoder UNet-like network, specifically PointTransformer [57], to re-capture semantic information present in the encoding. Important to its function is the contextualising modules ability to consider wider shape context, than either the UDF or segmentation decoder, which has repeatedly been shown as vital to capturing semantic information [39, 40]. This re-capturing of a wider scene context gives rise to the naming of our module. This is achieved through the PointTransformer's downsampling, interpolation, and upsampling performed across 5 different scales (similar to [40]).

The formulation of RangeUDF [50] which predicts the

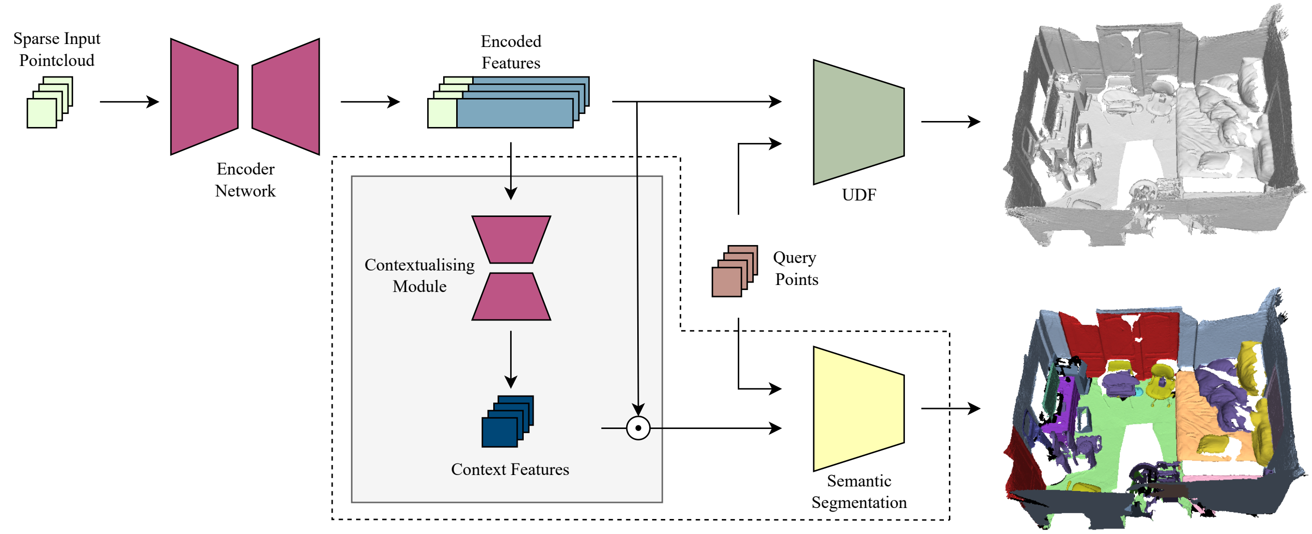
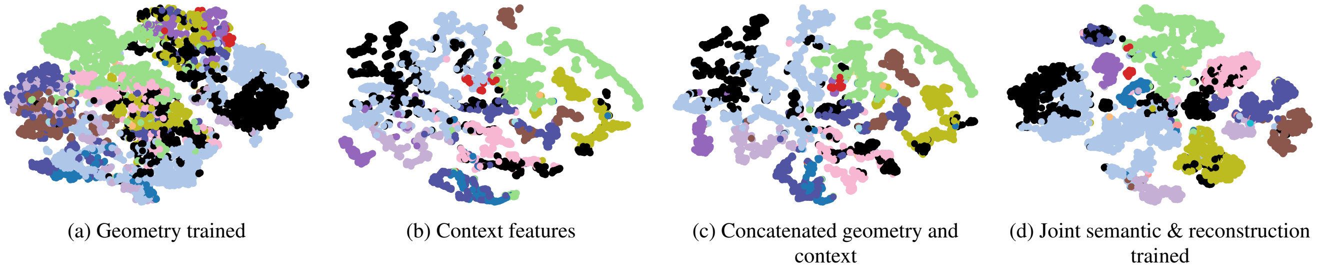


Figure 1. Our proposed contextualising module (light grey box) allows already trained implicit representations, to be fine-tuned (training only elements inside the dotted line) on semantic tasks without the need for the original training data. Learning a compact contextualising vector which is concatenated with the original encoding, our module allows full semantic performance to be recovered from encoders trained only on reconstruction tasks.



(a) Geometry trained

(b) Context features

(c) Concatenated geometry and context

(d) Joint semantic & reconstruction trained

Figure 2. t-SNE embeddings of the features of an encoding of a particular scene. The features trained for geomery tasks only, show poor separability according to semantic label. Our proposed contextualising module produces context features that are clearly more separable, and become even more so when combined with the existing features. Figure best viewed in colour.

semantic class, $s_i \in \mathbb{R}^C$ where C is the number of classes, of a given point $q_i \in Q$ as

$$s_i = f_{\text{sem}}(q_i|E_g) \tag{1}$$

Instead, our contextualising module, f_{ctx}, takes the fixed encoded features (trained on only the reconstruction task), $E_g \in \mathbb{R}^{M\times d}$, and predicts a set of context features, $E_c \in \mathbb{R}^{M\times l}$. We then concatenate the context features with the original encoded features to give the semantic features, $E_s \in \mathbb{R}^{M\times(d+l)}$, which we feed into the segmentation module alongside the query points, giving instead

$$s_i = f_{\text{sem}}(q_i|E_g \oplus f_{\text{ctx}}(E_g)) = f_{\text{sem}}(q_i|E_s) \tag{2}$$

where $\oplus$ represents concatenation in the feature dimension.

Despite the simplicity of our contextualising module and its implementation, our results demonstrate the performance improvements it provides to the semantic task are substantial. Our contextualising module is implemented as a substantially shrunk version of the PointTransformer, reducing the number of parameters from roughly 7.8 million to around 379,000. This is achieved through a reduction of the number of channels at each scale from $[32, 64, 128, 256, 512]$ to $[32, 32, 64, 64, 128]$ and reducing the number of "blocks" at each scale to 1.

During training, we use the L1 loss, with the same clamping as Chibane et al. [7], for supervising the reconstruction task, and the standard cross entropy loss for the segmentation task. Our method focuses on mainly on separately training each task, in which case, our loss contains only a single objective, avoiding the need to balance loss terms entirely. However, in the case of the joint training baseline, following [50] we use the uncertainty loss [22] to avoid manually tuning loss weightings between the semantic and reconstruction tasks.

4. Experiments

In this section, we cover details of the datasets and metrics used in our experiments, as well as the relevant details of our implementations and the resources used to perform our experiments.

4.1. Datasets & Metrics

We train and evaluate our method on three datasets: ScanNet [9], SceneNN [19], and 2D-3D-S [1], all three of which are captured using RGB-D cameras.

ScanNet The ScanNet dataset consists of 1613 scans of real-world rooms. The data is split into 1201 scans for training and 312 scans for validation with a further 100 scans held out for online benchmarks. Following Wang et al. [50], we use the validation set for testing as ground truth annotations for the test set are not publicly available. Semantic labels are provided for 40 classes, however following other methods [18, 27, 39, 40, 48, 50, 52], we train and test on only the 20 class subset used in the online benchmark.

2D-3D-S The 2D-3D-S dataset consists of 6 *very* large-scale indoor scans, capturing rooms, hallways and other educational and office like environments using an RGB-D The data is divided into a total of 271 rooms, divided into 6 “Areas” based on the scan they are contained in. Area 5 is split into two scans without a provided registration between them, preventing their use in the data preparation pipeline described below. Following Wang et al. [50], we use Areas 1-4 for training and Area 6 for testing. The semantic labels are provided for 13 classes.

SceneNN The SceneNN dataset consists of 76 indoor scans divided into 56 scenes for training and 20 scenes for testing [20]. Semantic labels are provided for the same 40 classes as ScanNet, where again we use the 20 class subset.

4.1.1 Data Preparation

We follow the same processing steps as Chibane et al. [7], normalising each scene’s mesh to a unit cube, and sampling 10k surface points (for the encoder input) and 100k off surface points for which we compute the distance to the closest point on the surface.

4.1.2 Metrics

When evaluating the reconstruction tasks, we use the standard Chamfer L1 & L2 distance measures (lower is better) as well as the F1-δ and F1-2δ score (higher is better). All CD-L1 values are reported $\times 10^{-2}$ and CD-L2 values $\times 10^{-4}$, and we set $\delta = 0.005$. For the segmentation task, we use mean Intersection-over-Union (higher is better) as well as mean F1-δ, which is calculated by determining the per-class F1-δ score then averaging over the classes.

Whilst we report both mF1-δ and mIOU for the semantic tasks, we suggest that more attention should be paid to the mF1-δ, as it better captures performance in the join task given that the IOU metric does not consider reconstruction performance at all. And the more extreme class imbalance present in the smaller SceneNN dataset can lead to greater instability and noise in this metric during evaluation.

At test time, like others [7, 50], we extract a mesh from the implicit representation, as well as semantic labels for 100k points on the surface of that mesh. We then compute the chamfer and F metrics against 100k points sampled directly from the ground truth meshes.

4.2. Implementation Details

We implement our work in PyTorch [36], and perform our experiments on 3 Nvidia RTX6000 GPUs and an Intel Xenon Gold 6226R CPU. We use the Adam optimiser with default parameters and a learning rate of 10^{-3} for all experiments, we use a batch size of 12, and set the dimension of the context features to 4. During training, we feed 10,240 points into the encoder, and 50k points to the UDF and segmentation decoder. For experiments on ScanNet, we train the model for 500 epochs. For both 2D-3D-S and SceneNN, we train for 1k epochs.

For the encoder network, we use the PointTransformer [57] network. To drastically speed up the evaluation of our experiments, we use the surface extraction algorithm from Zhou et al. [59] rather than Algorithm 1 from Chibane et al. [7].

5. Results

Comparing reconstruction-only trained features with our contextualised features We start with our experiments confirming that the findings of [8] apply to larger implicit representations. For our baseline, given no source code is publicly available, we use our implementation (Sec. 4.2) of RangeUDF [50] (the current SOTA method for joint reconstruction and segmentation), jointly training reconstruction alongside segmentation, as in the original paper. To confirm the findings, we train only the encoder network and UDF on the reconstruction task for a given dataset (2nd row Tabs. 1a to 1c), which we refer to as "Geometric Only". Then, freezing the encoder network and UDF (with these reconstruction only features), we train the segmentation decoder on the semantic labels of the same dataset, using the frozen encodings (3rd row Tabs. 1a to 1c), which we refer to as "Frozen Encoder". Its clear from the mIOU and mF1 scores that the frozen encodings are insufficient for reasonable quality segmentation results, with the frozen encodings giving less than half the performance of the baseline jointly trained model in the case of ScanNet. We also again train the segmentation decoder on the semantic labels, but provide no geometric supervision and do not freeze the encoder (4th row Tabs. 1a

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
Baseline	0.321	0.861	0.662	0.724
Geometric Only	0.302	0.884	-	-
Frozen Encoder	0.298	0.888	0.280	0.296
Unfrozen Encoder	0.768	0.506	0.587	0.744
Context Features	0.297	0.889	0.640	0.694

(a) ScanNet

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
Baseline	0.859	0.603	0.604	0.692
Geometric Only	0.832	0.633	-	-
Frozen Encoder	0.864	0.624	0.575	0.600
Unfrozen Encoder	1.18	0.430	0.532	0.657
Context Features	0.865	0.629	0.608	0.681

(b) SceneNN

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
Baseline	0.364	0.819	0.695	0.727
Geometric Only	0.389	0.822	-	-
Frozen Encoder	0.358	0.837	0.458	0.435
Unfrozen Encoder	0.971	0.31	0.359	0.734
Context Features	0.357	0.838	0.684	0.700

(c) 2D-3D-S

Table 1. Comparison of semantic segmentation and geometric reconstruction, on three datasets. The rows from top to bottom: joint training baseline, geometry reconstruction supervision only, semantic training on frozen encodings from geometry only, semantic training on frozen encodings with our contextualising module.

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
ScanNet Labels	0.297	0.889	0.639	0.694
SceneNN Labels	0.344	0.836	0.584	0.621
Stanford Labels	0.328	0.861	0.692	0.693

(a) ScanNet trained geometric features

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
ScanNet Labels	0.781	0.681	0.563	0.65
SceneNN Labels	0.793	0.631	0.562	0.648
Stanford Labels	0.740	0.626	0.5578	0.679

(b) SceneNN trained geometric features

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
ScanNet Labels	0.357	0.852	0.623	0.690
SceneNN Labels	0.379	0.806	0.611	0.648
Stanford Labels	0.357	0.838	0.684	0.700

(c) 2D-3D-S trained geometric features

	L1 (↓)	F1-δ (↑)	mF1-δ (↑)	mIOU (↑)
ScanNet Labels	0.299	0.887	0.634	0.690
SceneNN Labels	0.343	0.837	0.593	0.631
Stanford Labels	0.325	0.863	0.728	0.729

(d) Triad trained geometric features

Table 2. Cross training and validation using our contextualising module. For each table, we use the fixed feature encodings trained on reconstruction only for one dataset, and then train for segmenation with our contextualising module on each of the datasets. Triad represents the amalgamation of all three of the datasets.

to 1c), referred to as "Unfrozen Encoder". Whilst obviously unfair, this experiment demonstrates that although, as you would expect, semantic performance can be recovered, significant reconstruction capability is forgotten in the process.

Finally, we train the segmentation decoder with the frozen encodings combined with the context features produced by our contextualising module (5th row Tabs. 1a to 1c), which we refer to as "Context Features". The mIOU and mF1 scores show that our contextualising module allows nearly full performance on the segmentation task to be recovered.

Qualitative results are shown in Fig. 3, where the middle left shows the baseline results, and middle right shows the frozen encoder results, and the far right shows the results using the context features. Whilst bulk areas are easily segmented with the frozen encodings, there is significant confusion and error with smaller objects, and structurally similar surfaces are misclassified (*e.g*. in the 2nd row, the orange beam in the left corner and yellow blinds on the back wall are entirely missed by the fixed encoding).

Cross Training & Validation One of the key advantages of our method is that it allows for the separation of training for reconstruction tasks and semantic tasks without compromising on the performance of either. To evaluate this, we cross-train fixed reconstruction-only trained feature encodings (geometric only) with our contextualising module on each of the datasets (*i.e*. We train for reconstruction only on ScanNet and then for semantics on *etc*.).

We also train an additional set of fixed encodings on the amalgamation of the three datasets, which we refer to as the Triad dataset. To preserve train-test splits, the train split for Triad is sum of the three training splits, and likewise for the validation splits.

Our results in Tab. 2, specifically the mF1-δ score (which best describe the joint task performance), demonstrate that our method not only allows cross training between different datasets for reconstruction and semantic tasks, but *importantly*, our results in the 2nd and 3rd rows in Tab. 2d show that by leveraging the ability to train for reconstruction on larger datasets and then then semantics on a different smaller dataset, we can maintain the same semantic performance as a jointly trained baseline, whilst improving the quality of the reconstructions generated.

Whilst the triad dataset provides meaningful improvement to the semantic results (when the geometric features trained on the Triad are used to train the contextualising mod-

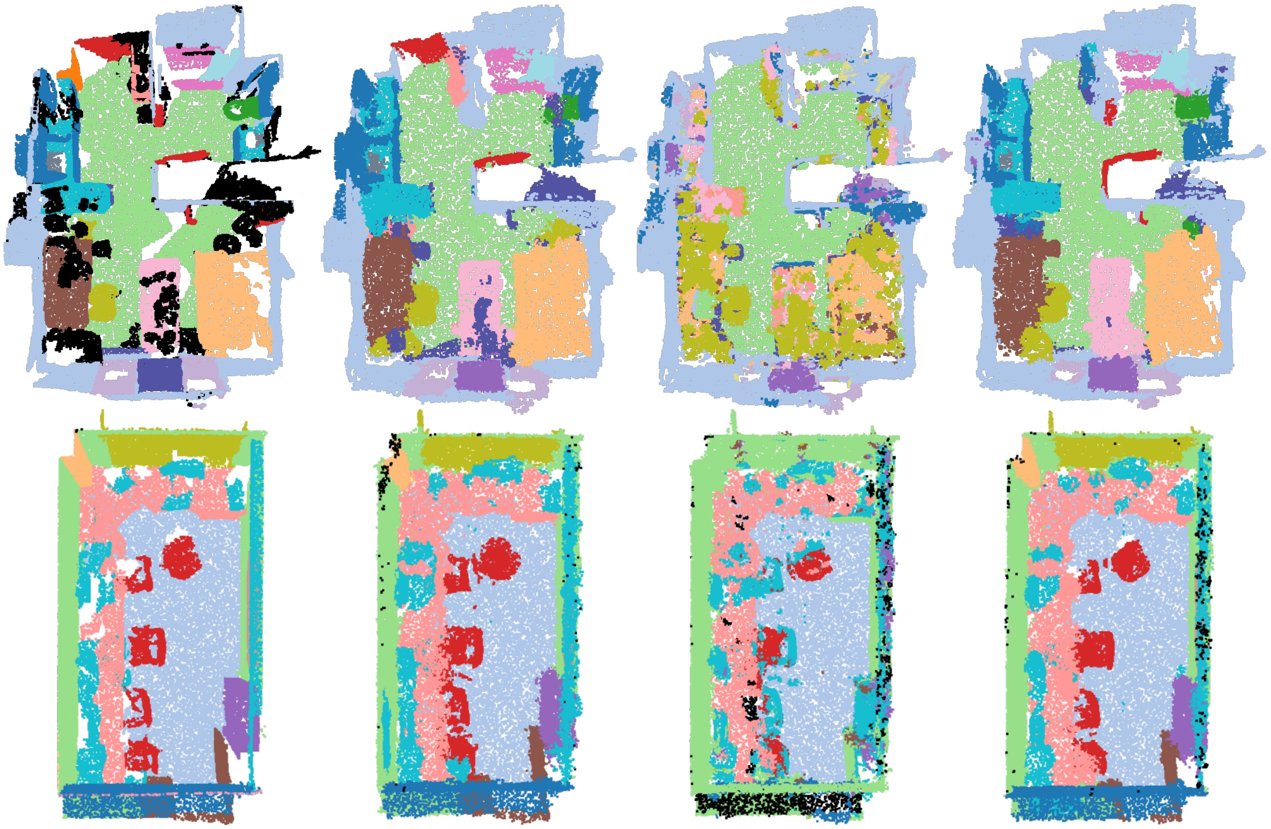


Figure 3. Qualitative comparison of segmentation and reconstruction on the ScanNet dataset. From left to right: Ground truth (semantics and geometry), jointly trained geometry and segmentation, segmentation training on frozen encodings, and finally segmentation training on frozen encodings using our contextualising module. The frozen encodings trained only for reconstruction seriously inhibit the network's performance on segmentation, misclassifying small objects, and entirely missing structurally similar classes (*e.g*. yellow curtain on back wall in 2nd row.)

ule on both the SceneNN and Stanford datasets) the same improvement is not seen for the results on ScanNet. We suggest that this arises from the relative sizes of the datasets, as the Triad dataset os only 30% larger than ScanNet, but is over ×5 larger than the Stanford dataset and over ×27 larger than SceneNN. As a result, whilst this means that a much broader feature space may be learned on the triad dataset compared to Stanford or SceneNN, on ScanNet this new feature space is arguably not meaningfully larger.

5.1. Ablations

To validate our design of the contextualising module and its compactness, we perform ablation experiments on the structure of both our contextualising module, as well as the context features we generate. For our ablation experiments, we use the 2D-3D-S dataset, all parameters are kept the same as in the above experiments except for the modifications described below.

In our experiments (Tab. 3), we evaluate the following:

	mF1-δ (↑)	mF1-2δ (↑)
MLP Context Module	0.490	0.568
Shallower network (3 scales)	0.644	0.745
More blocks	0.669	0.768
More channels	0.662	0.763
More blocks & channels	0.664	0.765
$l = 2$	0.606	0.706
$l = 1$	0.566	0.662
Ours	0.664	0.763

Table 3. Results of our ablation experiments on the 2D-3D-S dataset. For the experiments using more blocks we increase the number of blocks from $[1, 1, 1, 1, 1]$ to $[1, 2, 3, 5, 2]$. For the experiments using more channels, we increase the number of channels at each resolution scale from $[32, 32, 64, 64, 128]$ to $[32, 64, 128, 256, 512]$. l refers to the dimension of the context feature.

Contextual information To confirm that information from across the whole feature encodings for a given shape is vital to our contextualising module, rather than individual feature vectors, we implement our contextualising module as an MLP first, and second as a shallower version of the PointTransformer normally used. Our results show that the MLP provides little to no advantage over the raw fixed encodings, and that whilst the shallower PointTransformer recovers some of the performance, there is still a gap in performance compared to the baseline. These results demonstrate the importance of capturing contextual information across the whole encoding in our proposed module, and that simple re-projection of the fixed encodings is insufficient.

Compactness To show that our contextualising module is as compact as possible whilst maintaining performance, we evaluate the effects of increasing the size of the contextualising module, either by increasing the number of blocks at each scale, or by increasing the number of channels at each scale, or both simultaneously. Whilst these provide very marginal increase to performance, they both (particularly increasing the number of blocks) substantially increase the number of parameters in the contextualising module. We also demonstrate that further reducing the dimension of the context features below 4 harms the performance of the contextualising module. However, this specific parametrisation applies only to the datasets we use, and may be different for more complex or simpler datasets.

6. Limitations

Although Initial convergence of semantic segmentation performance when training the context module is faster than the joint training baseline, 90% of the performance with 17% of the training time, full convergence is not materially faster than the baseline. We suspect, however, this slowness arises from the segmentation module proposed in Wang et al. [50], as training the encoder used to generate the encoded features on a simple segmentation task converges substantially faster.

Ultimately, the main limitation of our approach is that it requires labelled data to train the semantic branch. However, as our approach separates the training of the reconstruction and semantic tasks, it is theoretically possible to extract meshes from the decoders at a coarse scale, and then manually label them to train the network for semantic tasks. There are also a number of weaknesses that arise from the design original RangeUDF [50], however these improvements would not necessarily represent any novelty, rather incremental improvements that would improve numerical performance, such as replacing their scalar attention module with vector attention or adding positional encoding [46] to the decoders.

7. Conclusion

In this work, we propose a novel approach to training implicit representations for downstream semantic tasks without needing access to the original training data or retraining the encoding network. We introduce our contextualising module that reveals hidden semantic information contained in the encodings of implicit representations trained only for geometric tasks. We demonstrate our contextualising module on the task of semantic segmentation and show that without it, the encoded features learnt by implicit representations for geometric tasks alone lack sufficient separability to provide meaningful results. Finally, we show that using our module, it becomes possible to leverage larger unlabelled datasets to pre-train implicit representations and then fine-tune on smaller labelled semantic datasets, achieving higher reconstruction performance than would be possible with only the smaller labelled datasets.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 2, 5

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 2

[3] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 2

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 2

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[7] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 4, 5

[8] Theo W Costain and Victor Adrian Prisacariu. Towards generalising neural implicit representations. *arXiv preprint arXiv:2101.12690*, 2021. 1, 2, 3, 5

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet:

Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 5

[10] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[12] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 3

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 2

[14] Benoît Guillard, Federico Stella, and Pascal Fua. MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks. In *Computer Vision – ECCV 2022*, pages 576–592. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science. 2

[15] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[17] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey. *arXiv preprint arXiv:2103.05423*, 2021. 3

[18] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 3, 5

[19] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 fourth international conference on 3D vision (3DV)*, pages 92–101. Ieee, 2016. 2, 5

[20] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 984–993, 2018. 3, 5

[21] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[22] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 4

[23] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *2020 International Conference on 3D Vision (3DV)*, pages 423–433. IEEE, 2020. 2

[24] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2

[25] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 3

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[27] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. Fpconv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2020. 3, 5

[28] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987. 2

[29] Luca De Luigi, Adriano Cardace, Riccardo Spezialetti, Pierluigi Zama Ramirez, Samuele Salti, and Luigi di Stefano. Deep learning on implicit neural representations of shapes. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[30] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 3

[32] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019. 2

[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2

[34] OpenAI. Gpt-4 technical report, 2023. 1

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous

signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2, 3

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[37] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2, 3

[38] Omid Poursaeed, Matthew Fisher, Noam Aigerman, and Vladimir G Kim. Coupling explicit and implicit surface representations for generative 3d modeling. In *European Conference on Computer Vision*, pages 667–683. Springer, 2020. 2

[39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3, 5

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 3, 5

[41] Zhongwei Qiu, Kai Qiu, Jianlong Fu, and Dongmei Fu. Dgcn: Dynamic graph convolutional network for efficient multi-person pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11924–11931, 2020.

[42] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018.

[43] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 3

[44] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32:1121–1132, 2019. 2

[45] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[46] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020. 2, 8

[47] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6504–6513, 2021. 2

[48] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 3, 5

[49] Suhani Vora*, Noha Radwan*, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *Transactions on Machine Learning Research*, 2022. https://openreview.net/forum?id=ggPhsYCsm9. 2

[50] Bing Wang, Zhengdi Yu, Bo Yang, Jie Qin, Toby Breckon, Ling Shao, Niki Trigoni, and Andrew Markham. Rangeudf: Semantic surface reconstruction from 3d point clouds. *arXiv preprint arXiv:2204.09138*, 2022. 1, 2, 3, 4, 5, 8

[51] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 197–213. Springer, 2022. 2

[52] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3, 5

[53] Haotian Xu, Ming Dong, and Zichun Zhong. Directionally convolutional networks for 3d shape segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2698–2707, 2017. 3

[54] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 2

[55] Xianghui Yang, Guosheng Lin, Zhenghao Chen, and Luping Zhou. Neural vector fields: Implicit representation by explicit learning. *arXiv preprint arXiv:2303.04341*, 2023. 1, 2

[56] Jianglong Ye, Yuntao Chen, Naiyan Wang, and Xiaolong Wang. Gifs: Neural implicit function for general shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12829–12839, 2022. 2

[57] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 3, 5

[58] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. *arXiv preprint arXiv:2103.15875*, 2021. 2

[59] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 5